

An Introduction to NuGet

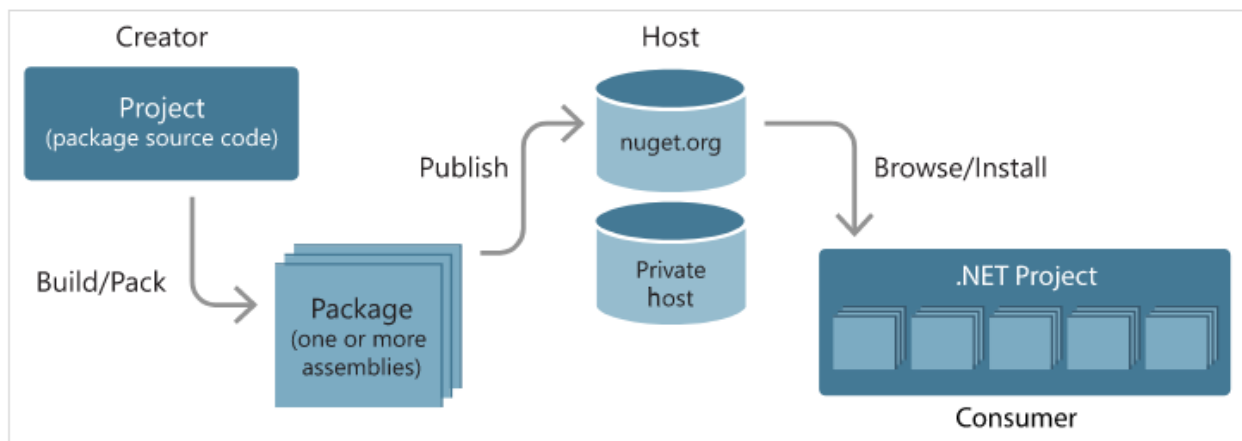
Overview

An essential tool for any modern development platform is a mechanism through which developers can create, share, and consume useful code. Often such code is bundled into "packages" that contain compiled code (**as DLLs**) along with other content needed in the projects that consume these packages.

For .NET, the Microsoft-supported mechanism for sharing code is NuGet (Pronunciation is similar to “New Get”), which defines how packages for .NET are created, hosted, and consumed and provides the tools for each of those roles. In short, NuGet is the package manager for .NET.

A NuGet package is a single ZIP file with the `.nupkg` extension containing compiled code (DLLs), other files related to that code, and a descriptive manifest that includes information like the package's version number.

The Flow of Packages Between Creators, Hosts, and Consumers

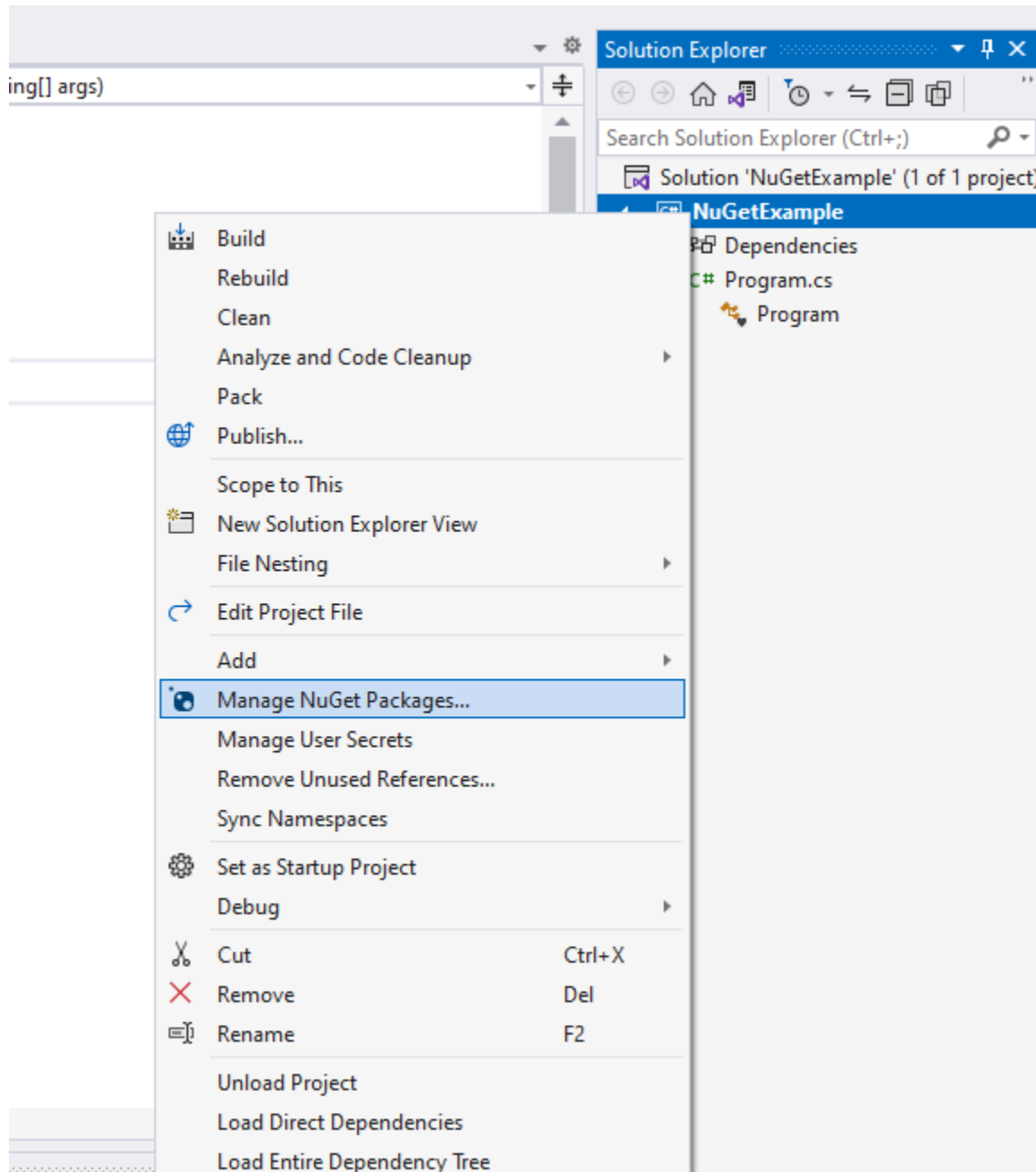


We need to know how to Browse/Install the NuGet package to our .NET Projects. Once the package is installed, we can use the classes and all the dependencies from that package.

How to Install NuGet to Your Project (Visual Studio 2022)

1. Right-click on the project you want to add NuGet packages and click **Manage NuGet Packages** (You can also access it under the project tab)

If you want to manage NuGet packages for your solution, Right-click on solution and then follow the same steps.



2. Search for the NuGet Package you want to add

2.1 You can find the package name, the author that developed it, the number of downloads, and the version number

(In this example, I'm going to download System.Data.SqlClient package, which will be used in tomorrow's session)

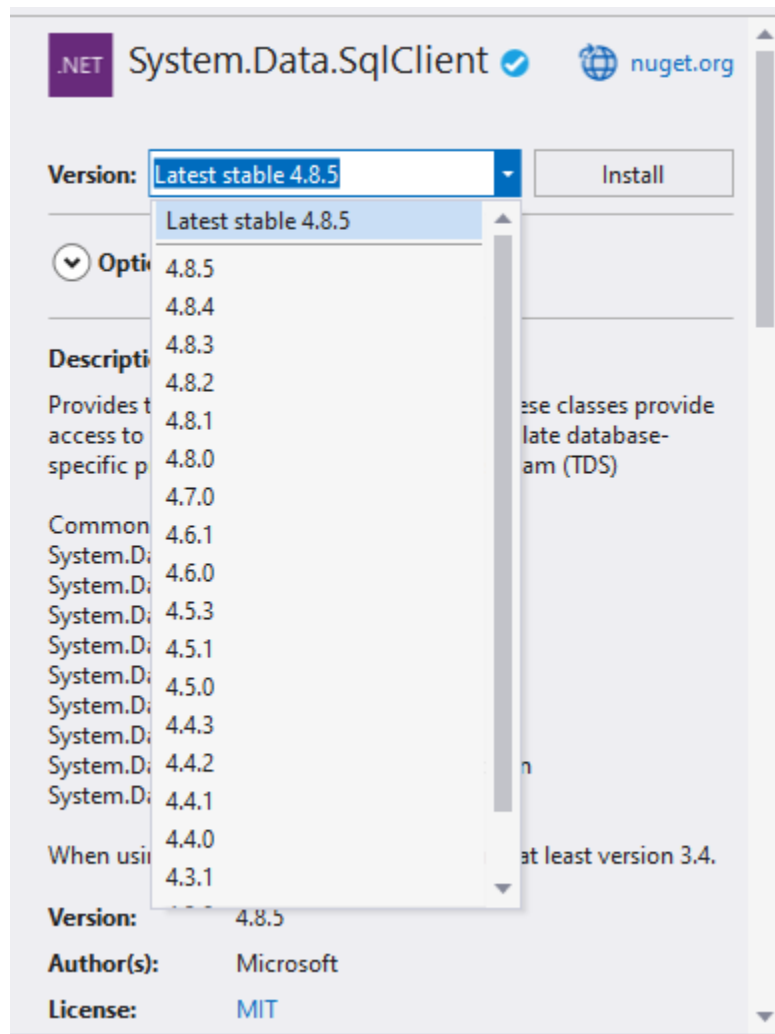
The screenshot shows the NuGet Package Manager interface with the search term 'sql'. The results list several packages, with 'System.Data.SqlClient' highlighted as the selected package. The interface includes tabs for 'Browse', 'Installed', and 'Updates', a search bar, and a checkbox for 'Include prerelease'.

Package Name	Author	Downloads	Version
Microsoft.EntityFrameworkCore.SqlServer	Microsoft	280M	7.0.1
Microsoft.Data.SqlClient	Microsoft	258M	5.0.1
System.Data.SqlClient	Microsoft	466M	4.8.5
SQLitePCLRaw.lib.e_sqlite3	Eric Sink	68.9M	2.1.3
SQLitePCLRaw.bundle_e_sqlite3	Eric Sink	64.2M	2.1.3
runtime.native.System.Data.SqlClient.sni	Microsoft	399M	4.7.0

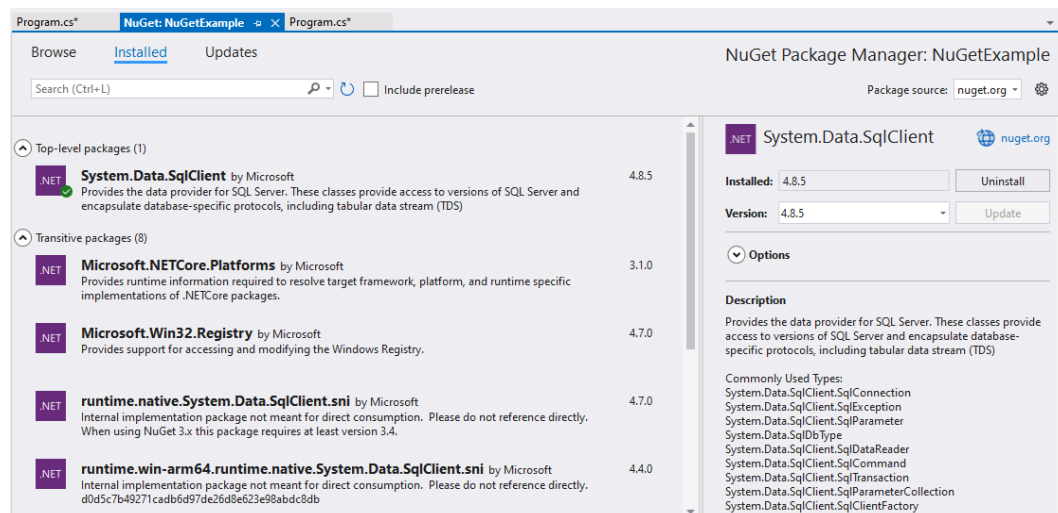
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

3. Select a stable version to install



4. Go to the installed tab, and you can find that the package has been installed in your project, you can also update or uninstall your NuGet package here



5. Now you can use the namespaces and classes within the NuGet package

The project w/ SQL NuGet package

```
using System.Data.SqlClient;

namespace NuGetExample
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Project with SQL NuGet package installed");
            SqlConnection conn = new SqlConnection();
        }
    }
}
```

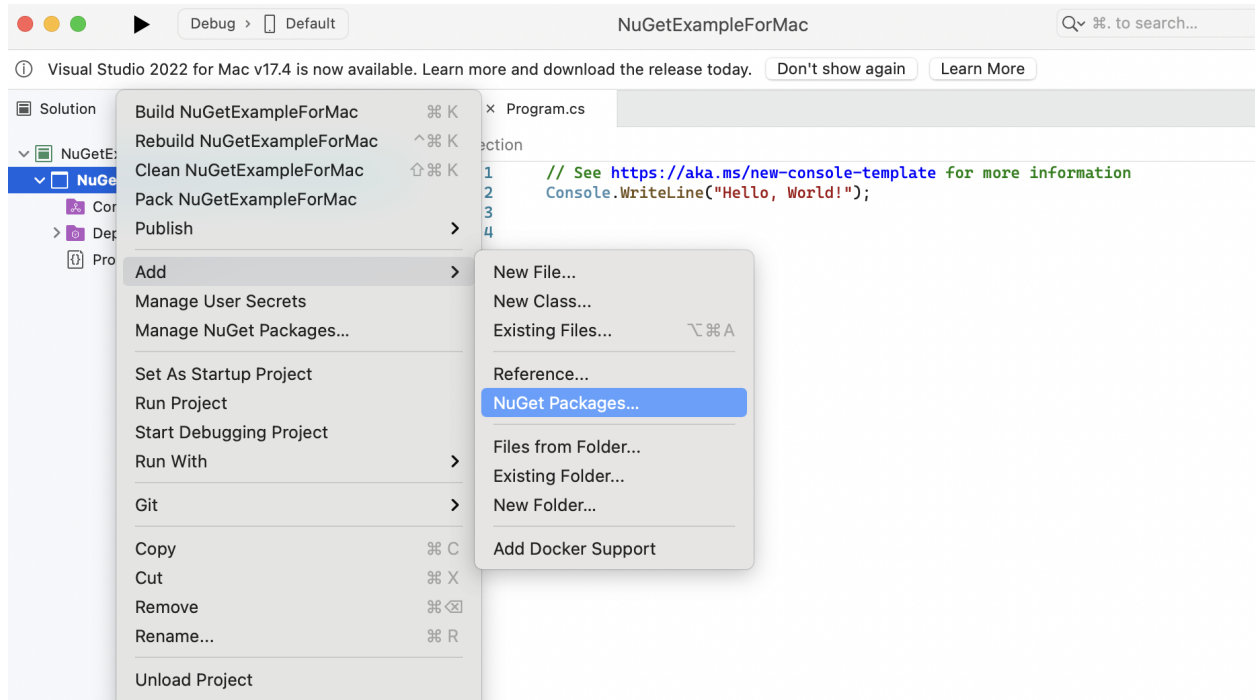
The project w/ out SQL NuGet package

```
using System.Data.SqlClient;

namespace NoNuGetInstalledExample
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Project with no SQL NuGet package installed");
            SqlConnection conn = new SqlConnection();
        }
    }
}
```

How to Install NuGet to Your Project (Visual Studio for Mac)

1. Similar to the steps in Windows



Other Resources

We can also download NuGet Package from the central repository using .NET CLI.

Central Repository: <https://www.nuget.org/>

Video References:

<https://www.youtube.com/playlist?list=PLdo4fOcmZ0oVLvfkFk8O9h6v2Dcdh2bh>